UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/724,285	11/26/2003	Debargha Mukherjee	200310818-1	1157

22879 7590 04/02/2009

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
PATEL, JAY P	
ART UNIT	**PAPER NUMBER**
2419	

NOTIFICATION DATE	DELIVERY MODE
04/02/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
jessica.l.fusek@hp.com

Office Action Summary	Application No.	Applicant(s)	
	10/724,285	MUKHERJEE ET AL.	
	Examiner	Art Unit	
	JAY P. PATEL	2419	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 November 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 1-24 are rejected under 35 U.S.C. 102(a) as being anticipated by the prior art non-patent literature submitted by the applicant and authored by Murkherjee et al. titled Structured Scalable Meta-formats (SSM) version 1.0 for content agnostic Digital Item Adaptation (herein referred to as Mukherjee).

1. In regards to claim 1, Mukherjee describes in section 5.1.4 on pages 24 and 25 (with making reference to figure 12 on page 24) a way to update a offset. The resource descriptor in figure 12 specifies a reference point in the bit-stream (R), the exact location in the bit-stream, the length in bits and endian type of where the value of an offset field is stored in the bit-stream, along with the actual numeric value (V) stored in this field (a descriptor comprising descriptor data comprising a reference point in the bit stream (R) and a numeric offset value (V) from the reference point in the bit-stream). The values R and V together provide the location of another point P in the bit-stream where P=R+V (said reference point and said numerical offset value having the ability to determine a pointer) (see figure 12 on page 24 and page 25, lines 4-9 under section 5.1.4).

Based on R and V, an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (see page 25, lines 12-19 under section 5.1.4).

First, if the field where an offset is stored or a part thereof is dropped (evaluating whether the offset value has been dropped from the bitstream)as part of adaptation, the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (all together anticipate performing, when the offset value has been dropped from the bitstream;)(see page 25, lines 16-21 under section 5.1.4).

The invalid pointers R or P, could be either moved up to the next valid byte (shifting a portion of descriptor data to a next byte when the offset value has been dropped from the bitstream), or moved down to the previous valid byte (shifting the portion of descriptor data to a previous byte when the offset value has been dropped from the bitstream), and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4). The third option zeros out the value V stored in the field when either R or P become invalid (setting the offset value to zero) (see page 25, lines 26-27 under section 5.1.4).

In regards to claim 2, The invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4). The third option is to zero out the value V stored in the field when either R or P become invalid (see page 25, lines 26-27 under section 5.1.4).

Furthermore, the semantics of the offset field in a given bitstream determines the most appropriate way to handle invalid pointers, but the descriptor should clearly mention the handling technique to be used (see page 25, lines 27-29 in section 5.1.4). An example is given on page 25, lines 30-34 in section 5.1.4.

In regards to claim 3, if the offset field indicates the length of R through P including R but excluding P with R<P, then both R and P should be moved up when invalid. Alternatively, if the offset field indicates the length of R through P excluding R but including P with R<P, then both R and P should be moved down when invalid (see page 25, lines 30-34 in section 5.1.4).

In regards to claim 4, one situation arises when either the byte at location R or the bytes at location P or both are removed (determining validity of at least one of reference point and pointer) as part of adaptation, however, the field where the offset is stored remains valid (before said evaluating)(see page 25, lines 16-21 under section 5.1.4).

In regards to claim 5, if the offset field indicates the length of R through P including R but excluding P with R<P, then both R and P should be moved up when invalid. Alternatively, if the offset field indicates the length of R through P excluding R

but including P with R<P, then both R and P should be moved down when invalid (see page 25, lines 30-34 in section 5.1.4).

In regards to claim 6, in a first situation, if the field where an offset is stored or a part thereof is dropped as part of adaptation, the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (see page 25, lines 16-21 under section 5.1.4).

In regards to claim 7, section 5 (page 23) is titled the XML contents.

In regards to claim 8, Figure 12 on page 24, clearly shows the location of the reference in the descriptor and the offset value in bitstream also in the descriptor.

2. In regards to claim 9, Mukherjee describes in section 5.1.4 on pages 24 and 25 (with making reference to figure 12 on page 24) a way to update a offset. The resource descriptor in figure 12 specifies a reference point in the bit-stream (R), the exact location in the bit-stream, the length in bits and endian type of where the value of an offset field is stored in the bit-stream, along with the actual numeric value (V) stored in this field (establishing at least one reference point (R), at least one numerical offset value (V)). The values R and V together provide the location of another point P in the bit-stream where P=R+V (and at least one pointer (P) in the descriptor associated with

the compressed bitstream) (see figure 12 on page 24 and page 25, lines 4-9 under section 5.1.4).

Based on R and V, an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (A method for updating offsets in a compressed bitstream upon dropping from the compressed bitstream using a descriptor) (see page 25, lines 12-19 under section 5.1.4).

First, if the field where an offset is stored or a part thereof is dropped as part of adaptation (evaluating whether dropped data from the compressed bitstream comprises at least a portion of the numerical offset value), the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (see page 25, lines 16-21 under section 5.1.4).

The invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (adjusting at least one of the reference point (R) and the pointer (P) when dropped data from the compressed bitstream comprises at least a portion of the numerical offset value) (see page 25, lines 23-26 under section 5.1.4). The third option is to zero out the value V stored in the field when either R or P become invalid (see page 25, lines 26-27 under section 5.1.4).

In regards to claim 10, Figure 12 on page 24, clearly shows the location of the reference in the descriptor and the offset value in bitstream also in the descriptor.

In regards to claim 11, the invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4).

In regards to claim 12, one situation arises when either the byte at location R or the bytes at location P or both are removed (determining validity of at least one of reference point and pointer) as part of adaptation, however, the field where the offset is stored remains valid(see page 25, lines 16-21 under section 5.1.4). This occurs before the evaluating step described in lines 23-26 on page 25.

In regards to claim 13, the invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4).

In regards to claim 14, section 5 (page 23) is titled the XML contents.

3. In regards to claim 15, Mukherjee describes in section 5.1.4 on pages 24 and 25 (with making reference to figure 12 on page 24) a way to update a offset. The resource descriptor in figure 12 specifies a reference point in the bit-stream (R), the exact location in the bit-stream, the length in bits and endian type of where the value of an offset field is stored in the bit-stream, along with the actual numeric value (V) stored in this field . The values R and V together provide the location of another point P in the

bit-stream where P=R+V (together anticipate, a method for updating offset values associated with a compressed resource bitstream after bitstream data drops using a descriptor comprising offset information) (see figure 12 on page 24 and page 25, lines 4-9 under section 5.1.4).

Based on R and V, an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (see page 25, lines 12-19 under section 5.1.4).

First, if the field where an offset is stored or a part thereof is dropped as part of adaptation (evaluating the compressed resource bitstream for dropping of offset information form the descriptor), the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (together anticipate repositioning offset information when the compressed resource bitstream includes dropped offset information from the descriptor) (see page 25, lines 16-21 under section 5.1.4).

The invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4).

The third option is to zero out the value V stored in the field when either R or P become invalid (see page 25, lines 26-27 under section 5.1.4).

In regards to claim 16, Figure 12 on page 24, clearly shows the location of the reference in the descriptor and the offset value in bitstream also in the descriptor.

In regards to claim 17, the first situation arises, if the field where an offset is stored or a part thereof is dropped as part of adaptation (evaluating the compressed resource bitstream for the dropping of the offset value), the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (together anticipate repositioning offset information when the compressed resource bitstream includes dropped offset information from the descriptor) (see page 25, lines 16-21 under section 5.1.4).

In regards to claim 18, the invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4). The third option is to zero out the value V stored in the field when either R or P become invalid (see page 25, lines 26-27 under section 5.1.4).

In regards to claim 19 based on R (reference point) and V (offset value), an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (see page 25, lines 12-15 under section 5.1.4).

In regards to claim 20, the invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4).

In regards to claim 21, section 5 (page 23) is titled the XML contents.

4. In regards to claim 22, Mukherjee describes in section 5.1.4 on pages 24 and 25 (with making reference to figure 12 on page 24) a way to update a offset. The resource descriptor in figure 12 specifies a reference point in the bit-stream (R), the exact location in the bit-stream, the length in bits and endian type of where the value of an offset field is stored in the bit-stream, along with the actual numeric value (V) stored in this field . The values R and V together provide the location of another point P in the bit-stream where P=R+V (together anticipate, a transcoder for updating offset values associated with a compressed resource bitstream after bitstream data drops using a descriptor comprising offset information) (see figure 12 on page 24 and page 25, lines 4-9 under section 5.1.4).

Based on R and V, an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (see page 25, lines 12-19 under section 5.1.4).

First, if the field where an offset is stored or a part thereof is dropped as part of adaptation (a compressed resource bitstream evaluator for evaluating the compressed resource bitstream for dropping of offset information form the descriptor), the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (together anticipate an offset information repositioner repositioning offset information when the compressed resource bitstream includes dropped offset information from the descriptor) (see page 25, lines 16-21 under section 5.1.4).

The invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4). The third option is to zero out the value V stored in the field when either R or P become invalid (see page 25, lines 26-27 under section 5.1.4).

5. In regards to claim 23, Mukherjee describes in section 5.1.4 on pages 24 and 25 (with making reference to figure 12 on page 24) a way to update a offset. The resource descriptor in figure 12 specifies a reference point in the bit-stream (R), the exact location in the bit-stream, the length in bits and endian type of where the value of an offset field is stored in the bit-stream, along with the actual numeric value (V) stored in this field . The values R and V together provide the location of another point P in the

bit-stream where P=R+V (together anticipate, A system for processing data in a compressed resource bitstream comprising, a transcoder for updating offset values associated with a compressed resource bitstream after bitstream data drops using a descriptor comprising offset information) (see figure 12 on page 24 and page 25, lines 4-9 under section 5.1.4).

Based on R and V, an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (see page 25, lines 12-19 under section 5.1.4).

First, if the field where an offset is stored or a part thereof is dropped as part of adaptation (a compressed resource bitstream evaluator for evaluating the compressed resource bitstream for dropping of offset information form the descriptor), the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid. For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (together anticipate an offset information repositioner repositioning offset information when the compressed resource bitstream includes dropped offset information from the descriptor) (see page 25, lines 16-21 under section 5.1.4).

The invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4).

The third option is to zero out the value V stored in the field when either R or P become invalid (see page 25, lines 26-27 under section 5.1.4).

6. In regards to claim 24, Mukherjee describes in section 5.1.4 on pages 24 and 25 (with making reference to figure 12 on page 24) a way to update a offset. The resource descriptor in figure 12 specifies a reference point in the bit-stream (R), the exact location in the bit-stream, the length in bits and endian type of where the value of an offset field is stored in the bit-stream, along with the actual numeric value (V) stored in this field (a descriptor comprising descriptor data comprising a reference point in the bit stream (R) and a numeric offset value (V) from the reference point in the bit-stream). The values R and V together provide the location of another point P in the bit-stream where P=R+V (said reference point and said numerical offset value having the ability to determine a pointer) (see figure 12 on page 24 and page 25, lines 4-9 under section 5.1.4).

Based on R and V, an adaptation engine can modify the field as and when bit-stream segments are dropped to update the value of the difference P-R (see page 25, lines 12-19 under section 5.1.4).

First, if the field where an offset is stored or a part thereof is dropped (determing whether the numerical offset value has been dropped) as part of adaptation, the entire entry corresponding to the field can be removed from the descriptor, because it does not need to be handled anymore. Next situation arises when either the byte at location R or the bytes at location P or both are removed as part of adaptation, however, the field where the offset is stored remains valid (determining whether at least one of the

pointer and the reference point have been removed when the numerical offset value has not been dropped and determining whether bytes proximate to the pointer and the reference point have been removed (since the bytes **at** P and R are evaluated)). For such situations, the descriptor should mention how the pointers are to be updated before the new V is computed and updated in the resource (updating the numerical offset value when at least one of the pointer and the reference point have been removed or bytes proximate to the pointer and the reference point have been removed)(see page 25, lines 16-21 under section 5.1.4).

The invalid pointers R or P, could be either moved up to the next valid byte, or moved down to the previous valid byte, and the result of the updated value V would be different based on which one is done (see page 25, lines 23-26 under section 5.1.4). The third option, zeros out the value V stored in the field when either R or P become invalid (setting the offset value to zero) (see page 25, lines 26-27 under section 5.1.4). All these anticipate the act of updating the offset.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JAY P. PATEL whose telephone number is (571)272-3086. The examiner can normally be reached on Mon.-Thurs.: 8:00 a.m.- 6:30 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Edan Orgad can be reached on (571)272-7884. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Edan Orgad/
Supervisory Patent Examiner, Art Unit 2419